

Andoni Rodríguez Díaz (ra_02@live.com.mx) / Ulises Juárez Martínez (ujuarez@ito-depi.edu.mx)

Objetivo general

Aplicar principios, técnicas y herramientas para la adaptación de software desde su representación en código intermedio mediante mecanismos de reflexión y soporte metodológico.

Introducción

La reflexión es la capacidad de un programa para analizar y modificar su propia estructura y estado en tiempo de ejecución. Esto permite obtener la información del sistema de manera más precisa. Los sistemas de software se someten al mantenimiento para prevenir o corregir errores, agregar nuevas funcionalidades y modificar aquellas que ya existen; sin embargo, es necesario detener el sistema para realizar los cambios, lo que causa que los propósitos del mismo no estén disponibles durante el proceso de mantenimiento, incluso implica el riesgo de generar errores de ejecución si no se conoce completamente la composición del sistema.

Resultados

Se diseñó un esquema para agregar las capacidades de adaptación a un sistema compilado, para conocer su estructura y reemplazar sus comportamientos sin alterar su representación en código intermedio (*bytecode*) ni detener su ejecución. Se hicieron uso de la reflexión de Java para obtener la información de cada clase y las extensiones de gestión de Java (*JMX*) para permitir el acceso a dicha información. Además, se trabajó con la reflexión para obtener la información de la invocación en cada método y en base a ello determinar las condiciones del cambio en el comportamiento.

Conclusiones

La reflexión permite conocer la información acerca de la estructura de un sistema o un componente, representa una alternativa para la adaptación de software, sin afectar su integridad o sus dependencias. La adaptación de software permite complementar a un sistema con funcionalidades provenientes de componentes externos que no se diseñaron para tal fin. La característica de análisis de la reflexión de Java permitió obtener la información necesaria de un sistema sin detener su ejecución, de esta forma el usuario tiene la capacidad para determinar en qué parte del programa realiza las modificaciones.

Referencias:

Forax, R., Duri, E., & Roussel, G. (2005). Reflection-based implementation of Java extensions: the double-dispatch use-case. *The Journal of Object Technology*,4(10).
 Dawson, D., Desmarais, R., Kienle, H. M., & Müller, H. A. (2008). Monitoring in adaptive systems using reflection. SEAMS '08.
 JMX Documentation. (n.d.). Obtenido de <http://www.oracle.com/technetwork/java/javase/tech/docs-jsp-135989.html>

